GROWL Scripts and Web Services

Rob Allan
Grid Technology Group
E-Science Centre
CCLRC Daresbury Laboratory

r.j.allan@dl.ac.uk

GROWL

Collaborative project (**JISC** VRE I programme) between CCLRC Daresbury Laboratory and the Universities of Cambridge and Lancaster.

Project Objectives: to produce a lightweight client-side Grid connection toolkit.

Primarily aimed at application developers as a Grid programming library.

Project competed end January 2007 but related work is continuing

Rob Allan
CCLRC Daresbury Laboratory

Outline

1. Background
2. Avoiding firewall problems
3. Use of GROWL Scripts for authentication, job submission and file management
4. GROWL + Web Services = the GROWL Grid Programming Library

Rob Allan
CCLRC Daresbury Laboratory

GROWL

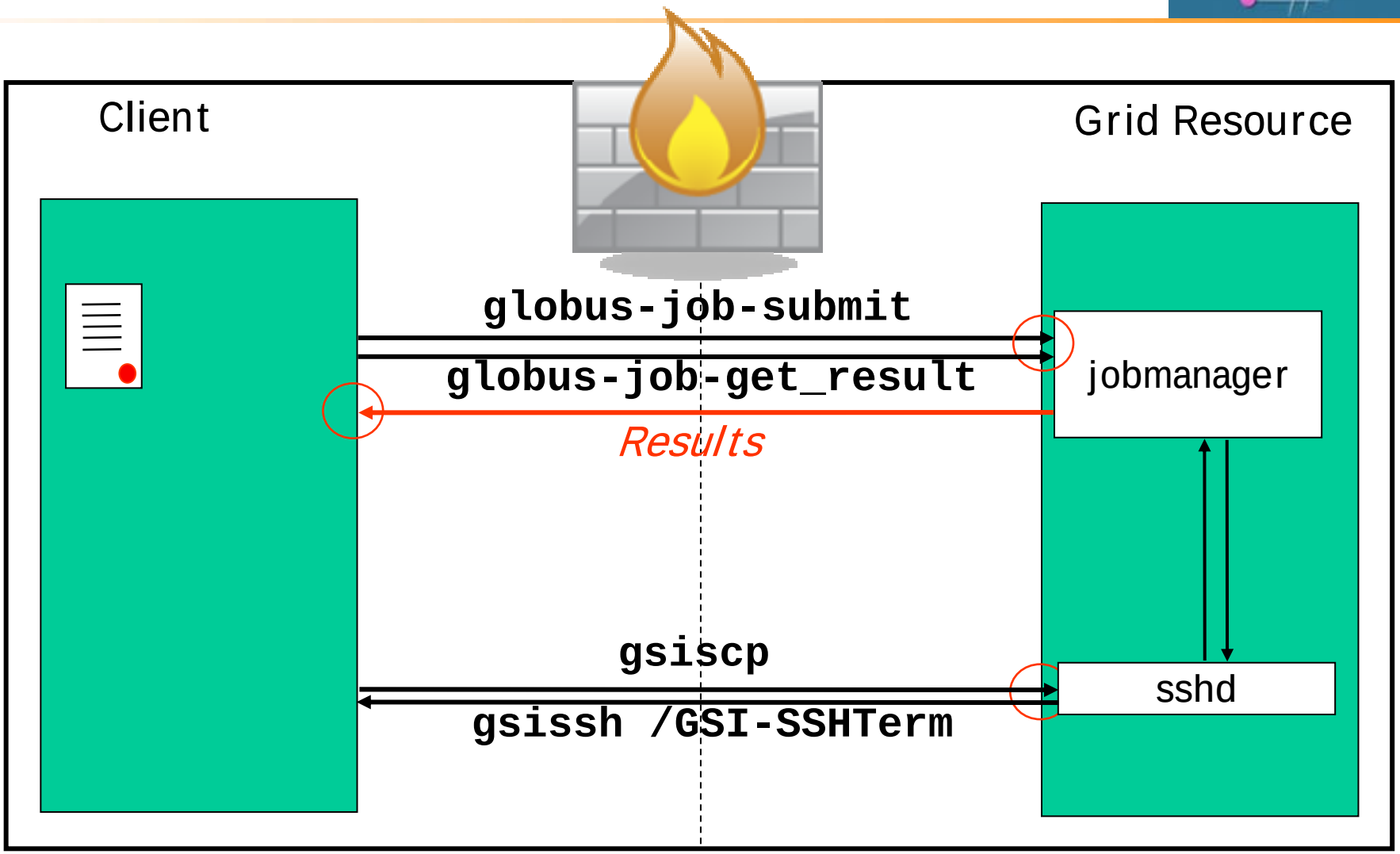
GROWL tries to address the three barriers that newcomers find when using the Grid for the first time:

1. Setting up the client-side middleware
2. Handling of certificates
3. **Job submission in the presence of firewalls**

We covered the first 2 of these in the previous NW-GRID training event, see http://www.nw-grid.ac.uk/?q=seminar/dares-250107

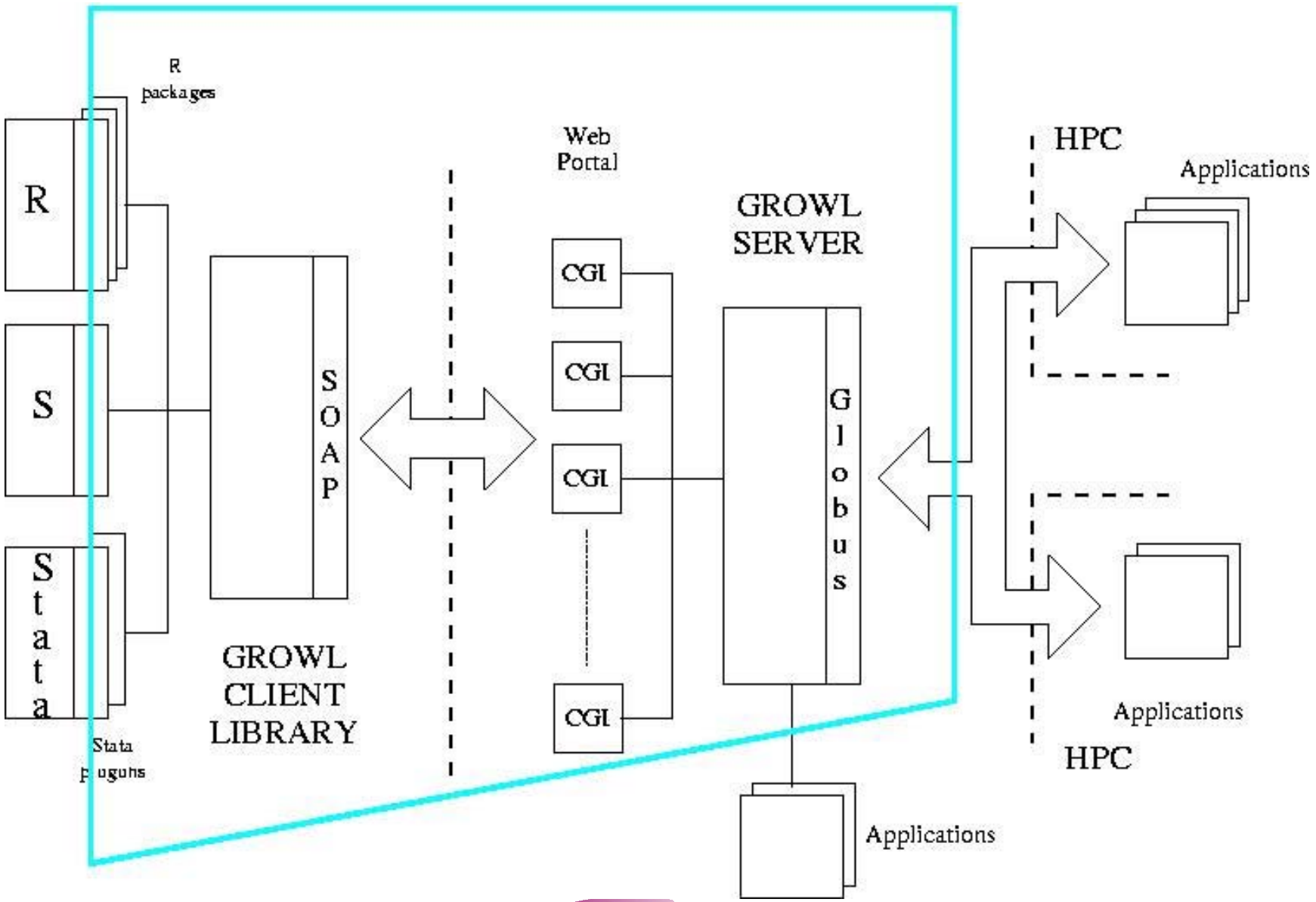
Rob Allan
CCLRC Daresbury Laboratory

Globus + Firewalls



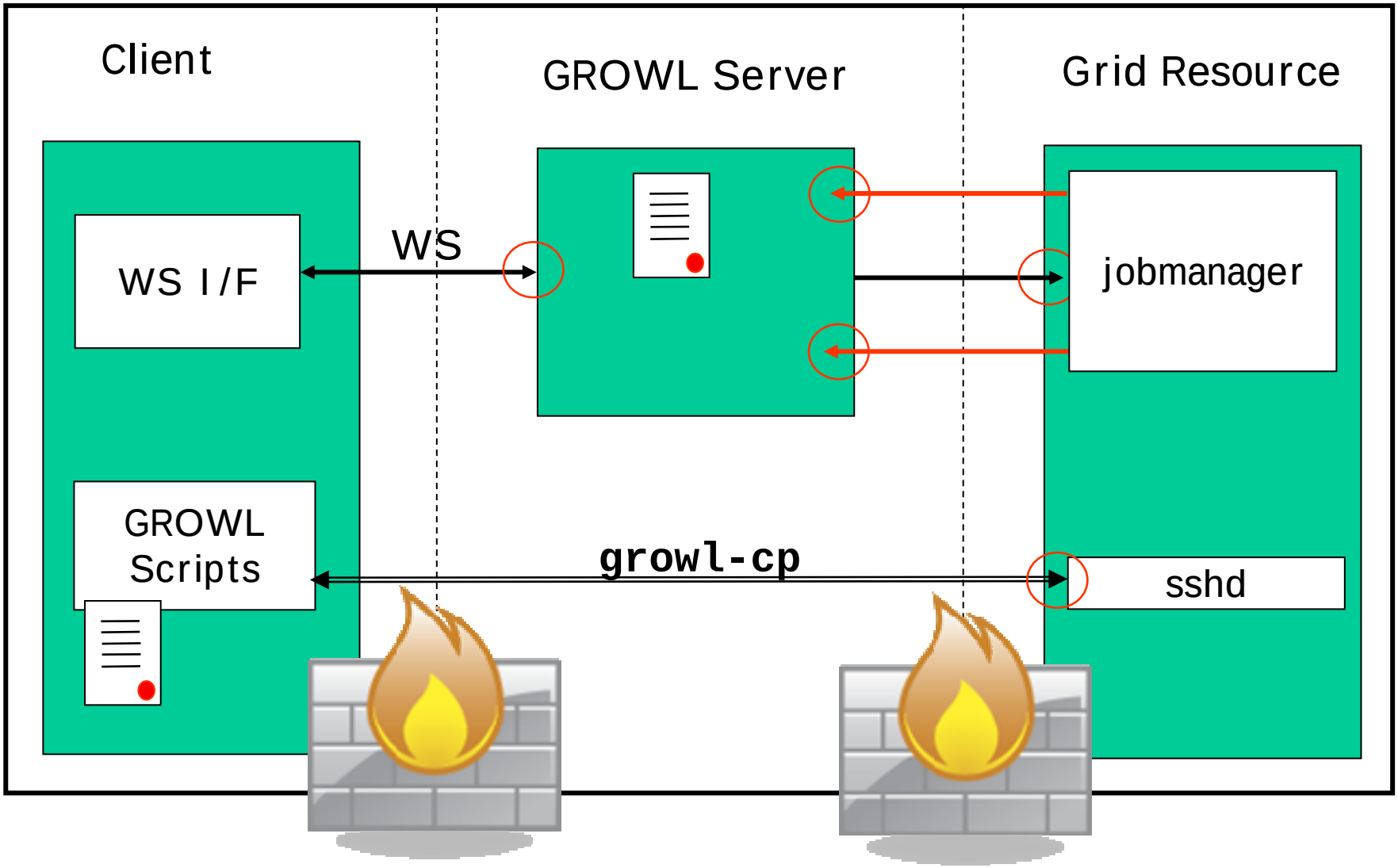
Rob Allan
CCLRC Daresbury Laboratory

How GROWL works



Rob Allan
CCLRC Daresbury Laboratory

GROWL + Firewalls (3-tier)



Rob Allan
CCLRC Daresbury Laboratory

GROWL Components

Simple GROWL (Little)

- Linux
- Window interface via Cygwin
- API uses only WS interfaces:
  - Authentication
  - Job Submission (Globus)
  - Simple indirect file transfer
  - Depends only on gSOAP
  - … (remote applications wrapped as WS – SRB, SABRE)
- [can be extended]

Full GROWL (Eagle)

- Linux Only
- Supporting scripts for
  - proxy management
  - remote filestore manipulation
- API has additional non WS components:
  - SRB client
  - Condor
  - Direct bulk file transfer
  - Other dependencies …

Rob Allan
CCLRC Daresbury Laboratory

GROWL Scripts – what you get:

- Certificate helper scripts
  - **`mk-cert`**
  - **`grow-info`**, **`grow-login`**, **`grow-logout`**
- VDT client installation of globus and MyProxy
  - **`grid-proxy-init`**, **`grid-proxy-info`**
  - **`globus-job-submit`**, **`globus-job-run`**
  - **`gsissh`**, **`gsiscp`**, **`openssl`**
  - **`myproxy-init`**, **`myproxy-info`**, **`myproxy-logon`**
- GROWL wrapper scripts
  - **`growl-submit`**, **`growl-status`**, **`growl-get-output`**
  - **`growl-sh`**, **`growl-cp`**, **`growl-mkdir`**, **`growl-rm`**,
  - **`growl-pwd`**, **`growl-which`**, **`growl-get-jobmanager`**

Rob Allan
CCLRC Daresbury Laboratory

Firstly

To add these various commands to your environment, add the following to your **.bashrc**, **.cshrc**, **.login**, **.profile** or equivalent.

```
# For /bin/sh or /bin/bash

$ . ~/Growl/setup.sh
```

```
# For /bin/csh or /bin/tcsh

$ source ~/Growl/setup.csh
```

Secondly

You can inter-mix Globus and GROWL, but

Before running any Globus commands or the GROWL wrapper scripts, you must have a valid local proxy certificate.

```
$ grid-proxy-init
Your identity: /C=UK/O=eScience/OU=CLRC/L=DL/CN=john kewley
Enter GRID pass phrase for this identity: <grid-password>
Creating proxy ..................................... Done
Your proxy is valid until: Tue Jan 23 05:19:34 2007
```


Rob Allan
CCLRC Daresbury Laboratory

growl-info

growl-info is a wrapper for **grid-cert-info**, **grid-proxy-info** and **myproxy-info**

```
$ growl-info
Certificate Information (including validity)
-------------------------------------------
subject= /C=UK/O=eScience/OU=CLRC/L=DL/CN=john kewley
notBefore=Jun 15 16:10:35 2006 GMT
notAfter=Jun 15 16:10:35 2007 GMT

Local proxy certificate(s)
--------------------------
subject  : /C=UK/O=eScience/OU=CLRC/L=DL/CN=john kewley
issuer   : /C=UK/O=eScience/OU=CLRC/L=DL/CN=john kewley
identity : /C=UK/O=eScience/OU=CLRC/L=DL/CN=john kewley
type     : Proxy draft (pre-RFC) compliant impersonation proxy
strength : 512 bits
path     : /tmp/x509up_u13445
timeleft : 11:57:19
```

Features of GROWL Scripts

1) Grid certificate handling is easier

2) Grid job submission

- Help with transparency - user shouldn't really need to know
  - Machine's jobmanager
  - Home directory location
  - Location in your path of executable
- Firewall problems minimised

3) File manipulation on client and remote resources

Rob Allan
CCLRC Daresbury Laboratory

grid-login

If you need to upload your certificate to MyProxy and generate a local proxy, growl-login is provided with the GROWL scripts:

```
$ grid-proxy-init
... <grid-password>

$ myproxy-init
Your identity: /C=UK/O=eScience/OU=CLRC/L=DL/CN=john kewley
Enter GRID pass phrase for this identity: <grid-password>
...
Enter MyProxy pass phrase: <myproxy-pass>
Verifying - Enter MyProxy pass phrase: <myproxy-pass>
```

```
$ growl-login
Password to protect MyProxy certificate: <myproxy-pass>
Enter GRID pass phrase for this id: <grid-password>
```

Running a grid job (1)

```
$ growl-submit dl1.nw-grid.ac.uk hostname
https://dl1.nw-grid.ac.uk:64010/792/116475/

$ growl-status https://dl1.nw-grid.ac.uk:64010/792/116475/
PENDING

$ growl-status https://dl1.nw-grid.ac.uk:64010/792/116475/
DONE

$ growl-get-output https://dl1.nw-grid.ac.uk:64010/792/116475/
comp023.nw-grid.ac.uk
```

```
$ growl-submit -c dl1.nw-grid.ac.uk hostname

$ growl-status
PENDING

$ growl-status
DONE

$ growl-get-output
comp021.nw-grid.ac.uk
```

Running a grid job (2)

Advantages of using GROWL wrapper scripts:

`growl-submit` :

- uses **`growl-get-jobmanager`** to obtain default parallel queue, rather than defaulting to **`jobmanager-fork`**
- uses **`growl-which`** to get full path of executable ensuring it is in your path

`growl-get-output` :

- uses **`gsissh`** to do remote retrieval, avoiding client firewall problem

Rob Allan
CCLRC Daresbury Laboratory

Remote filestore manipulation

Equivalents of many of the standard unix command tools are provided for remote filestore manipulation.

`growl-ls` : contents of directory

`growl-mkdir` : (sub)directory creation

`growl-rm` : file removal

`growl-cp` : remote file copying, including "3rd party"

`growl-which` : finds executable in your path

`growl-pwd` : prints your home directory on the grid resource

`growl-sh : gsissh wrapper (using default ports)`

They have an additional parameter of the remote resource on which they should run.

Remote file copying using `growl-cp`

`growl-cp` can be used to stage and retrieve files. The syntax is the same as that of scp. It can also be used for "3rd party" file transfers

```
$ growl-cp my_input_file.txt dl1.nw-grid.ac.uk:.
$ growl-cp dl1.nw-grid.ac.uk:my_output.txt .
$ growl-cp lv1.nw-grid.ac.uk:my_file.txt dl1.nw-grid.ac.uk:.
```

For 3rd party transfers to work, there has to be a route through all firewalls between the 2 remote resources in one direction or the other

Rob Allan
CCLRC Daresbury Laboratory

GROWL Web Services functionality

- Analogous interface to those we have seen on previous slides:
  - Authentication
  - Job submission
  - Remote file manipulation
  - Interfacing to SRB
  - Others indevelopment
- Wrappers being developed for access from R, Fortran, C and C++
  - Python and Perl later possibly
- Users at both Lancaster and Bristol using GROWL Web Services, although not yet available for widespread release

Rob Allan
CCLRC Daresbury Laboratory

GROWL Modules

- GROWL has a modular structure, so only the parts needed have to be installed
- GROWL uses several different servers for different purposes – in future it will use a registry to locate services fitting a functional description
- Current modules are:
  1. Authentication – test and renew proxy on the server(s)
  2. Jobs – submit interactive and batch jobs, upload stdin and download stdout and stderr via server
  3. Files – manage workspace on GROWL server (stdin, stdout, stderr and other named files in a “sandbox”)
  4. Transfer – transfer named files between client and/ or Grid resources. Uses GSI-SCP or Grid FTP where available
  5. SRB – manage virtual file store using SRB (uses an SRB server)
  6. Rcommands – meta-data management, some used internally
  7. Others being added – your requirements here…

Comparison with SAGA

SAGA: Simple API for Grid Applications is an OGF activity to define a Grid programming API.

- Job submission and management should be supported by the SAGA API.
- Resource discovery should be supported by the SAGA API.
- Data management should be supported by the SAGA API.
- Efficient data access should be supported by the SAGA API.
- Data replication should be supported by the SAGA API.
- Persistent storage of application specific information should be supported by the SAGA API.
- Streaming of data should be supported by the SAGA API.
- Asynchronous notification should be supported by the SAGA API.
- Support for messages on top of the streaming API should be considered by the SAGA API.
- Asynchronous notification should be supported by the SAGA API.
- Application level event generation and delivery should be supported by the SAGA API.
- Application steering should be supported by the SAGA API, but more use cases would be useful.
- GridRPC should be supported by the SAGA API.
- Access to data-bases does not currently require explicit support in the SAGA API.

Rob Allan
CCLRC Daresbury Laboratory

GROWL Library

GROWL_jobRun()
GROWL_jobSubmit()
GROWL_jobPoll()
GROWL_jobGetStatus()
GROWL_jobCancel()
GROWL_jobClean()
GROWL_jobGetStdoutAsString()
GROWL_jobGetStdoutAsFile()
GROWL_jobGetStderrAsFile()
GROWL_jobGetStderrAsString()
GROWL_jobPutFileToStdin()
GROWL_jobPutStringToStdin()

GROWL_getRemoteFile()
GROWL_getRemoteFileAsString()
GROWL_putRemoteFile()

Rob Allan
CCLRC Daresbury Laboratory

A Generic Example (in C)

```
// usual include files
…
// GROWL header file with function and server definitions
#include "growl.h“

int main(int argc, char **argv) {
  char *sessionId;
  int res;

// do authentication
  …
// run a job
  …
// get output
  …

  free(sessionId);
  return 0;
}
```


Rob Allan
CCLRC Daresbury Laboratory

Authentication

```
{
    printf("***** Testing old session key *****\n");
    char uid[] = "rja";
    char passPhrase[] = “my_passwd"; // preferably input from keyboard
    int ttl = 2;
    char oldId[] = "f6a6d3fe-1b36-4131-8dbe-dd29a5c483c5";

// check validity
    res = growl_testDelegation(oldId, &sessionId);
    if(strcmp(oldId,sessionId)) {
      strcpy(oldId, sessionId);
      printf("Got new session key %s\n", sessionId);
      printf("***** Need to renew session *****\n");

// get new proxy from MyProxy server onto GROWL server
      res = growl_getDelegation(uid, passPhrase, ttl, oldId, &sessionId);
    } else
      printf("***** Existing session still OK *****\n");
}
```

Running an interactive job

```
{
   char *response;

\\ set up parameters
   char *target="lv1.nw-grid.ac.uk";
   char *directory="/panfs/lv1/home/rja/SABRE";
   char *environment="";
   char *executable="/usr/local/sge6.0/bin/lx24-amd64/qsub";
   char *arguments="parallel.sh";

\\ submit the job
   res = growl_runClient(sessionId, target, directory, environment,
                         executable, arguments, &response);
   printf("Service returned: %s\n", response);
   free(response);
 }
```


Rob Allan
CCLRC Daresbury Laboratory

Getting the Output

```
{
   char *list;

\\ get stdout back from GROWL server
   res = growl_getOutput(sessionId, &list);
   if(res == 0) {
     printf("\n%s\n", list);
   } else
     printf("WS call failed\n");

\\ get stderrr back from GROWL server
   res = growl_getError(sessionId, &list);
   …
 }
```


Rob Allan
CCLRC Daresbury Laboratory

GROWL Web Site

Contains papers, presentations and other information.

Project Web site is being updated (JISC project funding finished end of January 2006). User manual/ tutorials are being added.

http://www.growl.org.uk/

Rob Allan
CCLRC Daresbury Laboratory